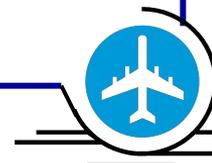


Введение в программирование

Лекция 8.

Примеры зачетных задач



Примеры зачетных задач

Билет включает **четыре задания** разных типов,
подобных заданиям 3_1 – 3_4.



Примеры зачетных задач

- **Задание 3_1.** Составить схему и трассировочную таблицу для данной программы.

Входной тест имеет вид: 24 60

```
/* Программа 3_1 */
#include <stdio.h>
void main (void)
{ int x, y;
  scanf ("%d %d", &x, &y);
  while (x > 0 && y > 0)
    if (x > y) x = x - y;
    else y = y - x;
  if (y == 0) printf ("%d", x);
  else printf ("%d", y);
}
```



Примеры зачетных задач

Решение задания 3_1.

Схема программы 3_1 приведена на рис. 8.1.

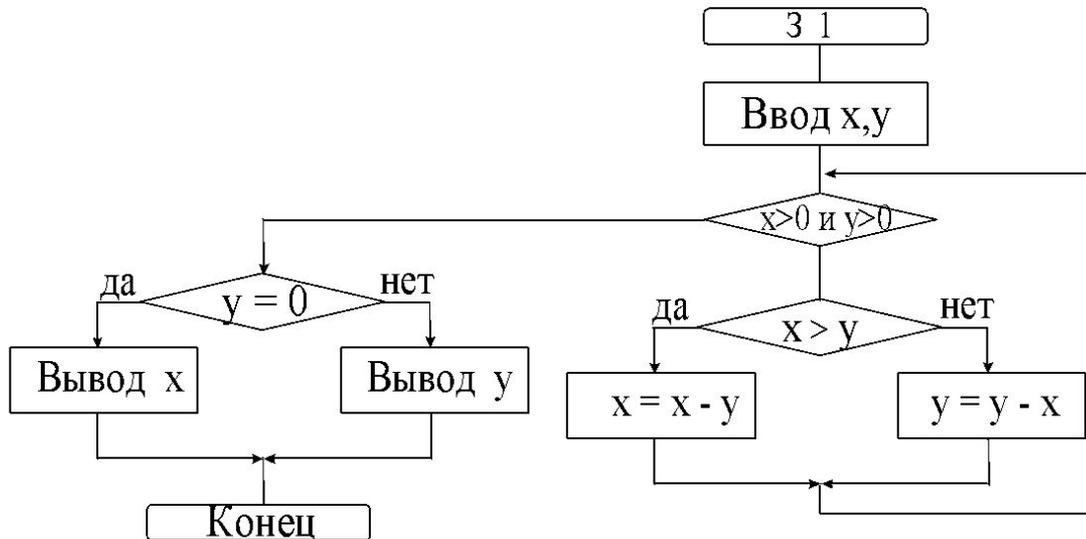


Рис. 8.1. Схема программы 3_1



Примеры зачетных задач

Табл.8.1. Трассировочная таблица программы З_1

x=	24						12				
y=	60		36		12				0		
x>0 && y>0		да		да		да		да		нет	
x>y		нет		нет		да		нет			
y==0											да

Вывод

Результат: 12



Примеры зачетных задач

Задание 3_2. Вычислить объем памяти для данных, определенных следующим образом:

```
int r[100];  
float x[10], z;  
char t[] = "KGTU"; char *s;
```

Решение задания 3_2.

объем = sizeof(r) + sizeof(x) + sizeof(z) + sizeof(t) +
sizeof(s) =

= (2*100 + 4*10 + 4 + 5 + 4) байт = 253 байт

Значением операции **sizeof (тип)** или

sizeof выражение является количество байтов,
необходимое для хранения в памяти данного типа
или данного выражения.



Примеры зачетных задач

- **Задание 3_3.** Дана последовательность из целых чисел - количество очков каждого из 500 участников соревнований.
Определить порядковые номера участников, набравших максимальное количество очков.
Составить схему и С-программу.

Решение задания 3_3 - программа 3_3 и
схема программы на рис. 8.2.



Схема программы на рис. 8.2.

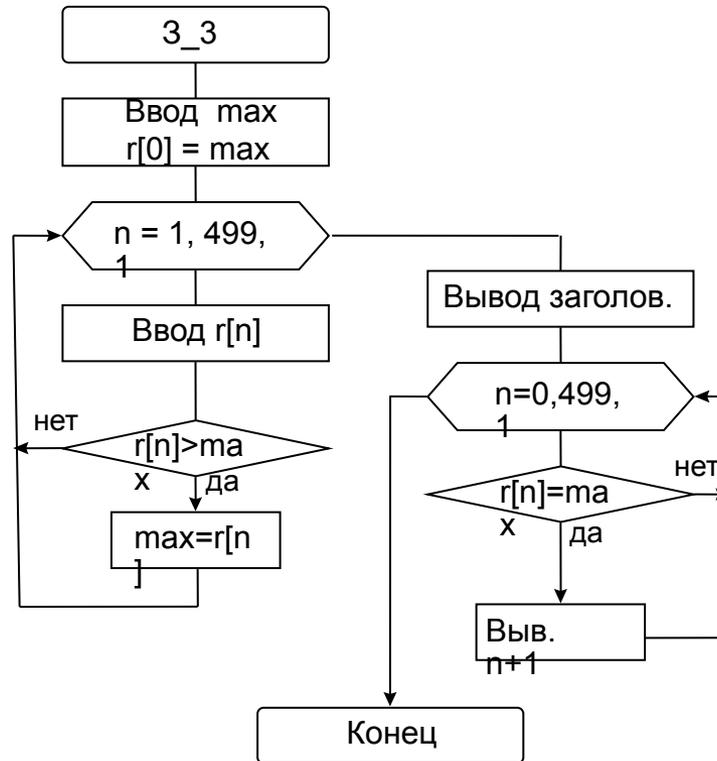


Рис. 8.2. Схема программы 3_3



Примеры зачетных задач

```
/* Программа 3_3. */
/* Получение номеров с максимальным результатом */
#include <stdio.h>
#define K 500 /* Количество участников */
void main (void)
{ int r[K]; /* результаты участников */
  int max; /* максимальный результат */
  int n; /* текущий номер участника */
/* Ввод и определение максимального результата */
  printf ("\nВведите результаты\n");
  scanf ("%d", &max); r[0] = max;
  for (n=1; n<K; n++)
  { scanf ("%d", &r[n]);
    if (r[n] > max) max = r[n];
  }
```



Примеры зачетных задач

```
/* Определение номеров победителей */
printf ("\nНомера победителей:\n");
for (n=0; n<K; n++)
  if (r[n] == max) printf (" %d", n+1);    /* 1..K */
}
```

- **Задание 3_4.** Составить подпрограмму подсчета количества повторений заданного символа в данной строке. Привести пример ее вызова.
- **Решение А.** Результат - возвращаемое значение, S – массив символов строки.



Примеры зачетных задач

```
/* Программа 3_4а */
/* Функция: количество повторений символа С в строке S */
int kol_simv (int c, char s[])
{ int kol;          /* Значение функции */
  int i;           /* Индекс текущего символа строки */
  kol = 0;
  for (i = 0; s[i] != '\0'; i++)
    if (s[i] == c) kol++;
  return kol;
}
```



Примеры зачетных задач

- **Решение Б.** Результат - выходной параметр kol, s - адрес строки.

```
/* Программа 3_4б */
/*Подпрограмма: kol = кол-во повторений символа c в строке s */
void p_kol_simv (int c, char *s, int *kol)
{ char *i;      /* Адрес текущего символа строки */
  *kol = 0;
  for (i = s; *i != '\0'; i++)
    if (*i == c) (*kol)++;
  return;
}
```



Примеры зачетных задач

- Пример вызова подпрограмм `kol_simv` и `p_kol_simv`.

```
void main (void)
{ int k;
  p_kol_simv ('a', "клавиатура", &k);
  printf ("\nКол. символов 'a' в слове ' клавиатура '
= %d = %d ", kol_simv ('a', " клавиатура "), k);
}
```

Результаты тестирования:

Кол. символов 'a' в слове " клавиатура " = 3 = 3



Примеры зачетных задач

```
/* Драйвер для проверки работы подпрограммы          */  
   /* удаления заданного символа в данной строке      */  
void main (void)  
{ int st[80]; int k=0;  
  gets(st);  
  k=ud('ы', st);  
  puts(st);  
  printf("\n Количество удаленных букв = %d", k);  
}
```

